

---

# Database in R

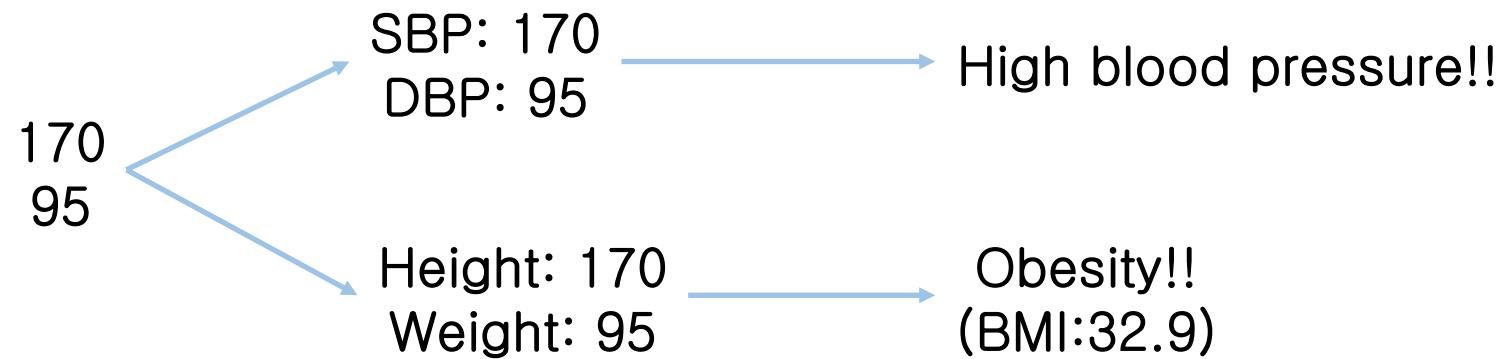
---

연세대학교 예방의학교실 유승찬  
Department of Preventive Medicine, Seng Chan You



# 데이터베이스와 데이터베이스 시스템

- 데이터 vs 정보 vs 지식
  - 데이터 (data): 관찰의 결과로 나타난 실제 값
  - 정보 (information): 데이터에 의미를 부여한 것
  - 지식 (knowledge): 사물이나 현상에 대한 이해



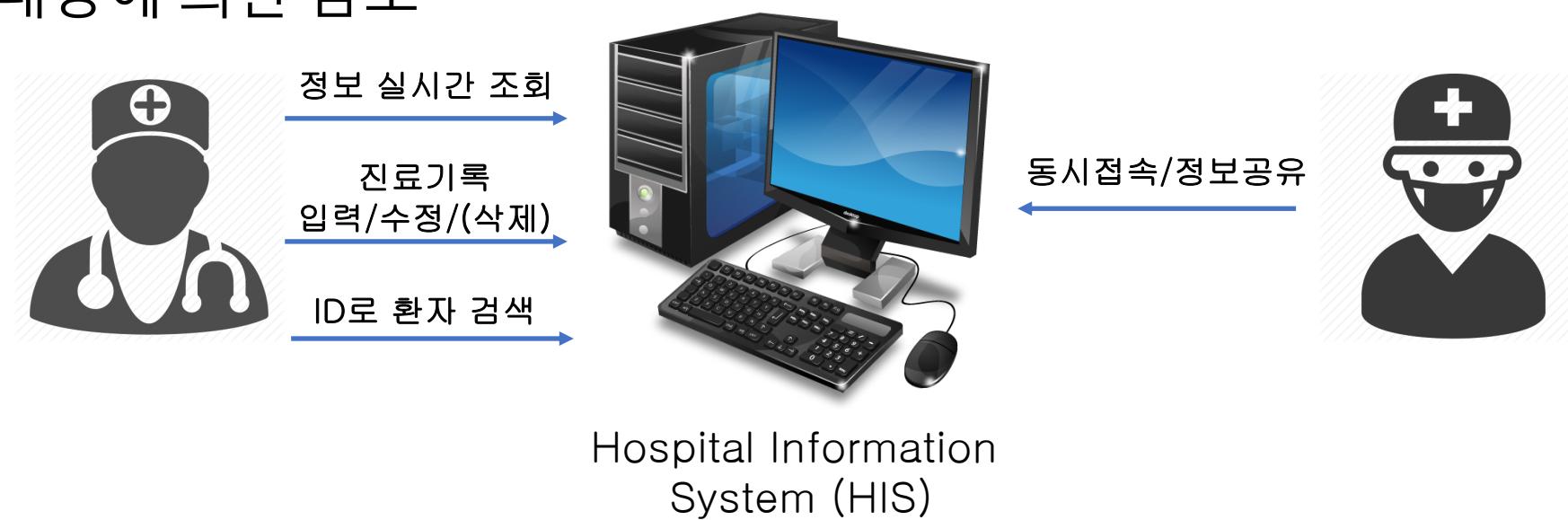
# 데이터베이스와 데이터베이스 시스템

- 데이터베이스 (Database)
  - 체계화된 데이터의 모임이다
  - 즉, 작성된 목록으로써 여러 응용 시스템들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터들의 묶음
- 어디에서 사용되고 있나?
  - 학교, 은행, 병원, 쇼핑몰, 도서관.....

# 데이터베이스와 데이터베이스 시스템

- 데이터베이스의 특징

- 실시간 접근성
- 지속적인 변화
- 동시 공유
- 내용에 의한 참조



# 데이터베이스와 데이터베이스 시스템

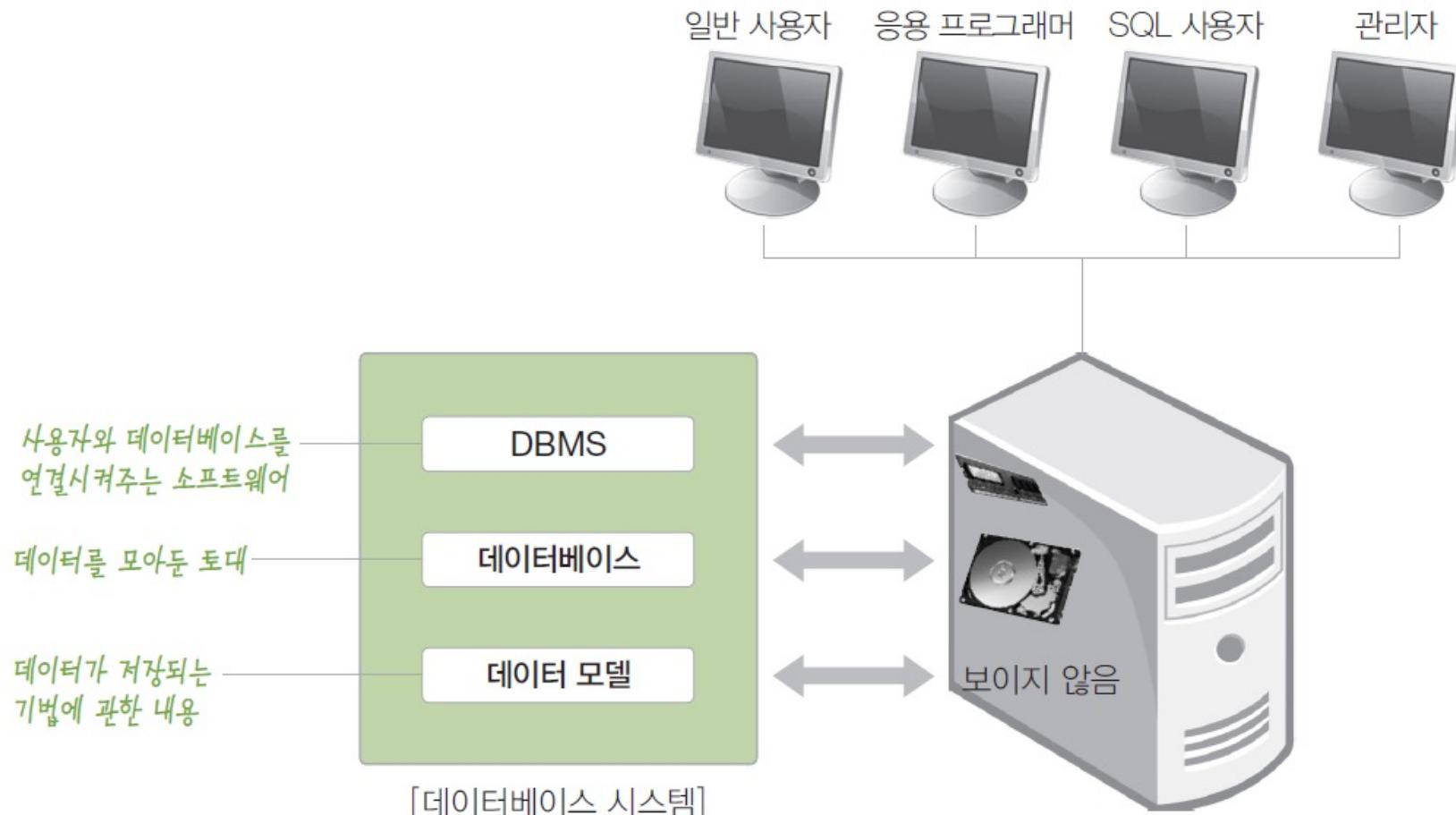
- 데이터베이스의 개념
  - 통합된 데이터(Integrated data)
    - 여러 사용자가 이용하는 데이터를 통합관리하여 중복을 최소화함으로써 데이터간의 불일치를 없앰
  - 저장된 데이터(Stored data)
    - 문서가 아닌 컴퓨터의 저장장치에 저장
  - 운영 데이터(Operational data)
    - 특정 업무를 위해 사용되는 데이터
  - 공용 데이터(shared data)
    - 여러 사용자가 공동으로 사용하는 데이터

# 데이터베이스와 데이터베이스 시스템

- 데이터베이스 관리 시스템 (DBMS, Database Management System)
  - 다수의 사용자들이 데이터베이스 내의 데이터를 접근할 수 있도록 해주는 소프트웨어 도구의 집합
  - DBMS은 사용자 또는 다른 프로그램의 요구를 처리하고 적절히 응답하여 데이터를 사용할 수 있도록 해준다.
- DBMS 기능
  - 정의 : 데이터에 대한 형식, 구조, 제약조건들을 명세
  - 구축 : DBMS가 관리하는 기억 장치에 데이터를 저장
  - 조작 : 질의, 캡션, 보고서 생성 기능 등
  - 공유 : 여러 사용자와 프로그램이 데이터베이스에 동시에 접근
  - 보호 : 하드웨어나 소프트웨어의 오동작 또는 권한이 없는 악의적인 접근으로부터 시스템을 보호
  - 유지보수 : 시간이 지남에 따라 변화하는 요구사항을 반영

# 데이터베이스와 데이터베이스 시스템

- 데이터베이스 시스템의 개념

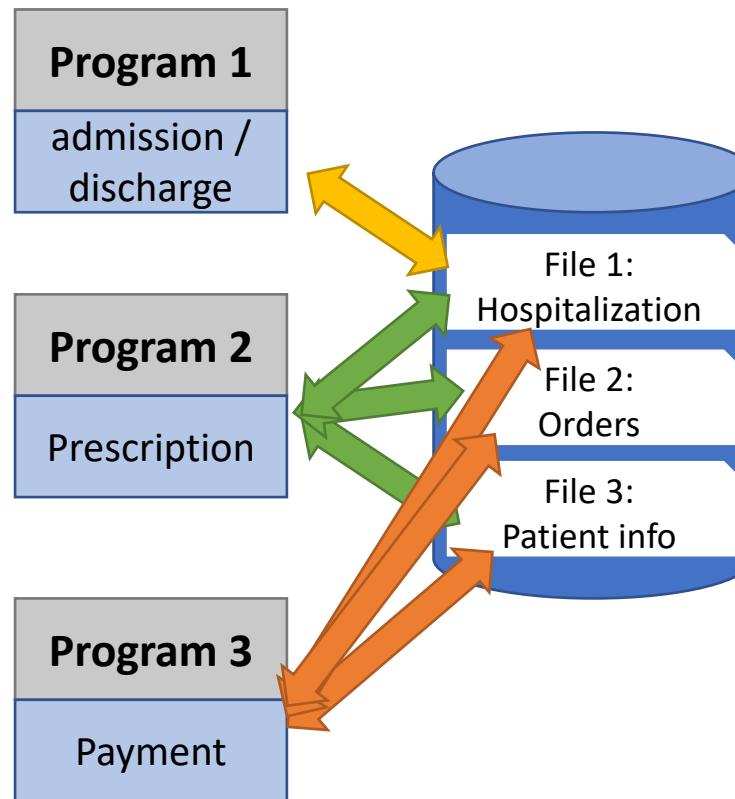


# 파일 시스템과 데이터베이스 시스템

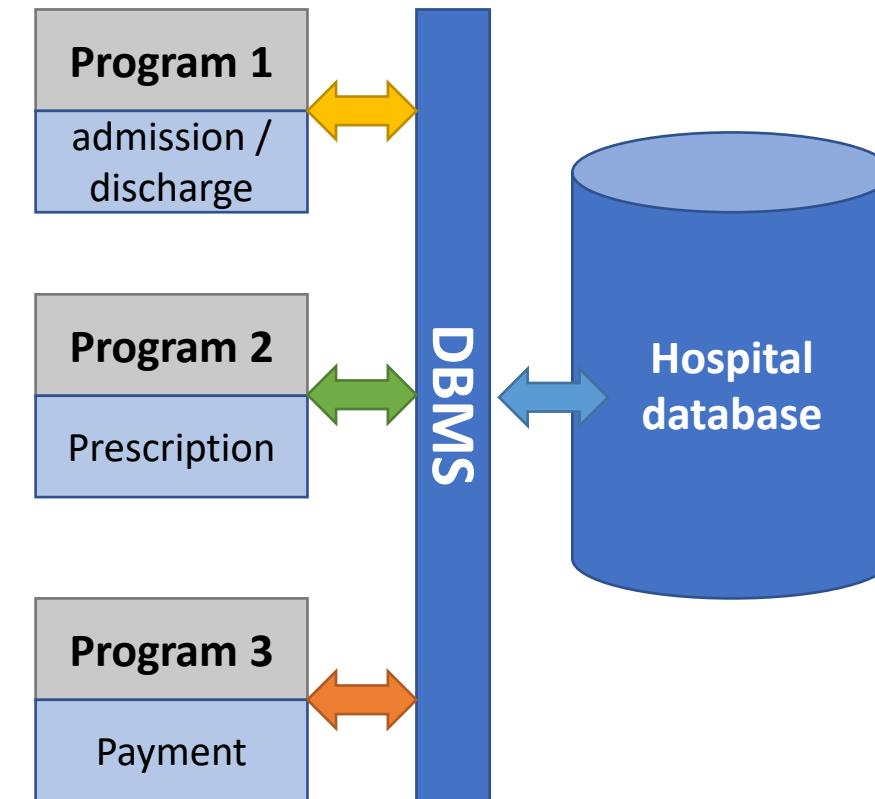
특성	파일 시스템	데이터베이스 시스템
데이터에 대한 접근	물리적 접근	물리적 접근 & 논리적 접근
동시 사용	동일한 파일을 두 개 이상의 프로그램이 동시에 접근할 수 없음	동일한 데이터를 다수 사용자가 동시에 접근 가능
구조화 및 중복성	비구조적. 중복성과 유지보수 비용 높음.	구조화되어 있음. 중복성과 유지보수 비용 낮음
데이터 공유	특정 프로그램만 접근	접근 권한이 있는 모든 프로그램이 공유
데이터 접근	미리 작성된 프로그램을 통해서만 접근	다양한 질의어를 사용하여 융통성 있는 접근 가능
통합성	각 응용 프로그램마다 파일이 따로 존재	데이터가 중복을 배제하면서 통합되어 있음

# 파일 시스템과 데이터베이스 시스템

파일 시스템



데이터베이스 시스템



# 파일 시스템과 데이터베이스 시스템

- 파일 시스템
  - Small system
  - Relatively cheap
  - Simple structure
  - Redundant data
  - Chance of inconsistency
  - No security
  - Isolated data
  - Often single user
- 데이터베이스 시스템
  - Large system
  - Expensive
  - Complex structure
  - Reduced redundancy
  - Consistent
  - Security
  - Data can be shared
  - Multiple users

# 파일 시스템과 데이터베이스 시스템

## ❖ 데이터베이스 관리 시스템 도입에 따른 장점

- ① 데이터의 중복과 불일치 감소
- ② 사용자에게 더 나은 서비스 제공
- ③ 시스템의 융통성이 향상됨
- ④ 시스템 개발 및 유지 비용 감소
- ⑤ 표준화 시행이 용이
- ⑥ 보안이 향상됨
- ⑦ 데이터 무결성이 향상됨
- ⑧ 조직체의 요구 사항을 파악하여 조정할 수 있음
- ⑨ 시스템의 고장으로부터 데이터베이스를 복구할 수 있음
- ⑩ 데이터베이스의 공유와 동시 접근이 가능함

# 데이터 모델

- 데이터 모델
  - 데이터베이스 구조의 근간
  - 데이터베이스 설계 과정에서 데이터의 논리적인 구조를 표현하기 위해 사용하는 도구
- 데이터 모델 종류
  - 계층형 데이터 모델
  - 네트워크형 데이터 모델
  - 관계형 데이터 모델
  - 객체지향형 데이터 모델

# 관계형 데이터베이스

- 릴레이션(테이블)으로 구성된 데이터 항목들의 집합
- 릴레이션(테이블)은 행과 열로 구성
- 테이블은 DB에서 표현하는 하나의 엔티티에 관한 정보를 저장
  - 개체(엔티티, entity): 인간의 개념 또는 정보의 세계에서 의미있는 하나의 정보 단위
- 예시: DB2, Oracle, MySQL, MS SQL Server, Access, PostgreSQL, BigQuery 등

테이블/ 릴레이션			
ID	Sex	Birthday	Ethnicity
0000001	1	1958-02-12	1
0000002	2	1962-07-31	1
0000003	2	1977-12-23	Null
0000004	1	1980-02-13	1

행/record/  
tuple

열/속성/attribute

# 관계형 데이터베이스

- 용어
  - 스키마, 인스턴스
  - 도메인
  - 차수
  - 카디널리티
  - 널값
  - 키, 후보키, 기본키, 대체키, 외래키

# 관계형 데이터베이스

- 스키마 (schema) / 내포
  - 릴레이션의 기본적인 구조를 정의
- 인스턴스 (instance) / 외연
  - 정의된 스키마에 따라 테이블에 실제로 저장되는 데이터의 집합

ID	Sex	Birthday	Ethnicity
0000001	1	1958-02-12	1
0000002	2	1962-07-31	1
0000003	2	1977-12-23	Null
0000004	1	1980-02-13	1

스키마

인스턴스

# 관계형 데이터베이스

- 도메인 (domain)
  - All the values which a data element may contain

ID	Sex	Birthday	Ethnicity
0000001	1	1958-02-12	1
0000002	2	1962-07-31	1
0000003	2	1977-12-23	Null
0000004	1	1980-02-13	1

정수  
0000000  
~9999999

Boolean  
1 or 2

날짜  
1800/01/01  
~2999/12/31

정수  
or Null

# 관계형 데이터 베이스

- 차수(degree)
  - 속성의 개수
- 카디널리티 (cardinality)
  - 튜플의 수

차수 = 4

카디널리티  
= 4

ID	Sex	Birthday	Ethnicity
0000001	1	1958-02-12	1
0000002	2	1962-07-31	1
0000003	2	1977-12-23	Null
0000004	1	1980-02-13	1

# 관계형 데이터 베이스

- 널값 (null)
  - 값이 없음. 존재하지 않는 값. missing value.
  - 관측된 값인 0과는 다른 (N/A와는?)
  - $3*0=0$
  - $3*Null=?$

ID	Sex	Birthday	Ethnicity
0000001	1	1958-02-12	1
0000002	2	1962-07-31	1
0000003	2	1977-12-23	Null
0000004	1	1980-02-13	1

0000003번  
환자의 인종은  
정보가 '존재하지 않음'

# 관계형 데이터베이스

- 키, 후보키, 기본키, 대체키, 외래키
  - 키 (key): 각 투플을 고유하게 식별할 수 있는 하나 이상의 애트리뷰트들의 모임
  - 후보키 (candidate key): 각 투플을 고유하게 식별하는 최소한의 애트리뷰트들의 모임 (두개 이상으로 이루어질 경우: 복합키)
  - **기본키 (primary key)**: 후보 키가 두 개 이상 있을 경우, 이들 중에서 하나를 기본 키로 선정함
  - 대체키 (alternate key): 기본키가 아닌 후보키
  - **외래키 (foreign key)**: 다른 릴레이션의 기본 키를 참조하는 애트리뷰트

<Patient>

ID	Sex	Birthday	Ethnicity
0000001	1	1958-02-12	1
0000002	2	1962-07-31	1
0000003	2	1977-12-23	Null
0000004	1	1980-02-13	1

기본키

<Laboratory>

ID	labdate	labname	value
0000001	2009-04-05	serum potassium	4.10
0000002	2008-10-03	serum potassium	3.80
0000002	2008-10-03	serum calcium	9.10
0000003	2009-04-01	serum potassium	4.60
0000004	2008-11-14	serum potassium	4.70
0000004	2008-11-14	serum magnesium	0.90

외래키

# 관계형 데이터베이스

- 도메인 무결성 제약조건
  - 도메인 제약(domain constraint)이라고도 함
  - 릴레이션 내의 투플들이 각 속성의 도메인에 지정된 값만을 가져야 한다는 조건
  - SQL 문에서 데이터 형식(type), 널(null/not null), 기본 값(default), 체크(check) 등을 사용하여 지정할 수 있음
- 개체 무결성 제약조건
  - 기본키 제약(primary key constraint)이라고도 함
  - 기본키는 NULL 값을 가져서는 안되며 릴레이션 내에 오직 하나의 값만 존재해야 한다는 조건
- 참조 무결성 제약조건
  - 외래키 제약(foreign key constraint)이라고도 함
  - 자식 릴레이션의 외래키는 부모 릴레이션의 기본키와 도메인이 동일해야 함
  - 자식 릴레이션의 값이 변경될 때 부모 릴레이션의 제약을 받음

# SQL (Structured Query Language)

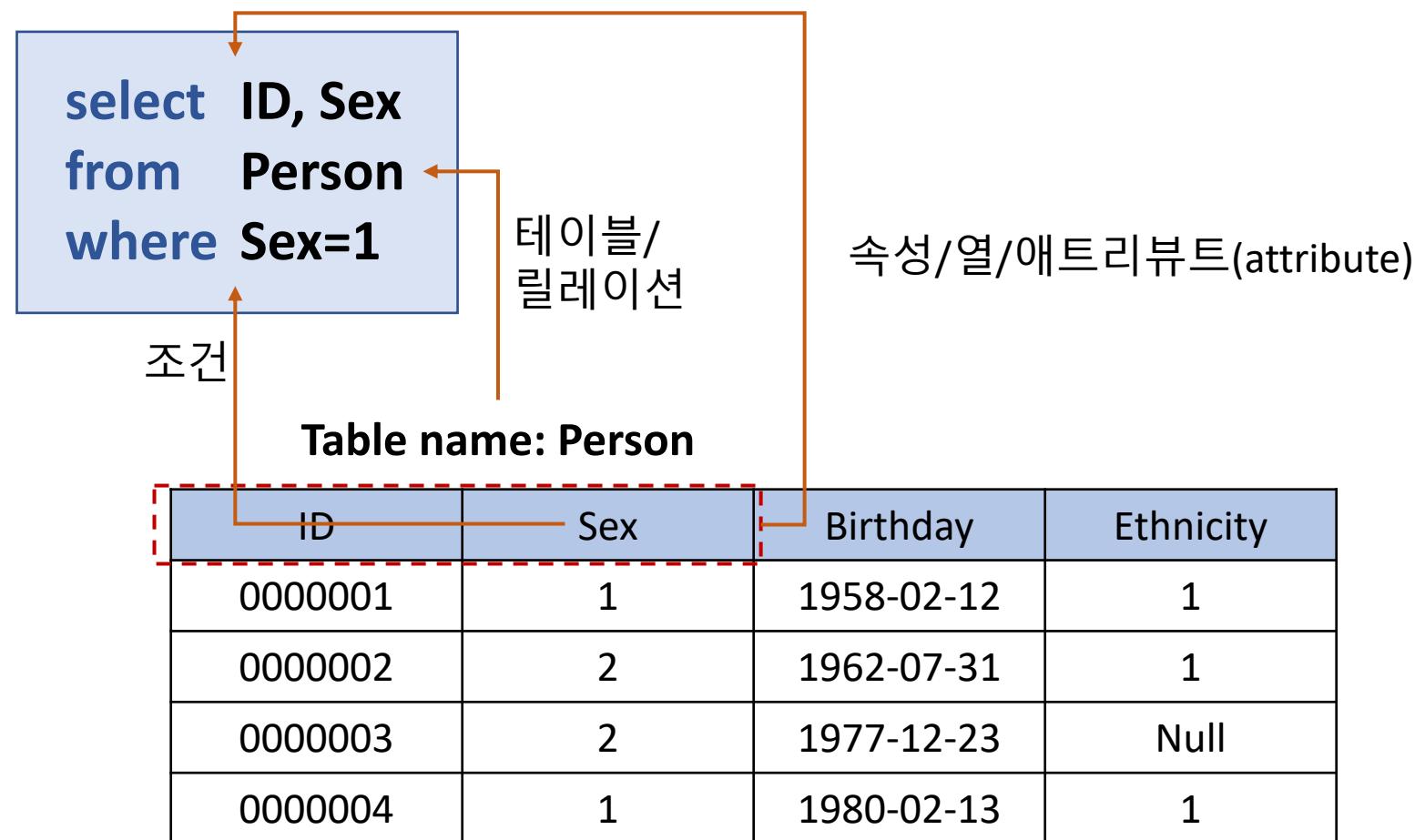
- 관계형 데이터베이스의 조작과 관리에 사용되는 데이터베이스 질의용 언어
- IBM의 DB2, 마이크로소프트의 Access와 SQL Server를 비롯하여 Oracle, Sybase, Informix 등에서 구조화 질의어로 널리 사용
- 3가지 기능
  - 데이터 정의 기능
  - 데이터 조작 기능
  - 데이터 제어 기능

# SQL (Structured Query Language)

- 데이터 정의어
  - 데이터 저장 구조, 데이터 접근 방법, 데이터 형식 등 데이터베이스를 구축하거나 수정할 때 사용하는 언어
  - create/drop/alter
- 데이터 조작어
  - 데이터베이스에 저장된 데이터를 검색, 수정, 삽입, 삭제할 때 사용하는 언어
  - select/insert/delete/update
- 데이터 제어어
  - 데이터를 보호하고 관리하는 언어
  - 데이터베이스의 무결성 유지, 보안 및 접근 제어, 시스템 장애로부터의 복구, 병행 수행 제어 기능 등을 수행

# 데이터 조작어 – select/from

- select 문의 구성 요소



# 데이터 조작어 – where

술어	연산자	예
비교	=, <>, <, <=, >, >=	QTc > 450
범위	BETWEEN	QTc BETWEEN 400 AND 500
집합	IN, NOT IN	personid IN (12345, 23456, 34567)
패턴	LIKE	diagnosis LIKE '%diabetes%'
NULL	IS NULL, IS NOT NULL	PR IS NULL
복합조건	AND, OR, NOT	(Sex = 1) AND (QTc > 450)



와일드 문자	의미	사용 예
+	문자열을 연결	'골프' + '바이블' : '골프 바이블'
%	0개 이상의 문자열과 일치	'%축구%' : 축구를 포함하는 문자열
[ ]	1개의 문자와 일치	'[0-5]%' : 0-5 사이 숫자로 시작하는 문자열
[^]	1개의 문자와 불일치	'[^0-5]%' : 0-5 사이 숫자로 시작하지 않는 문자열
_	특정 위치의 1개의 문자와 일치	'_구%' : 두 번째 위치에 '구'가 들어가는 문자열

# 조인(Join)

- 두 개 이상의 테이블을 연결
  - ex) 환자들의 진단내역을 진단명과 함께 추출

`select * from sample_Diagnosis`

personid	diagdate	diagcode	diaglocalcode	diagdept
118568	2001-07-29 16:39:00.000	R61.9	DC9940	I
391675	1999-11-18 15:43:00.000	C22.0B	DC13882	I
236390	2007-12-27 00:00:00.000	K21.0	DC13744	O
236390	2007-10-04 00:00:00.000	E03.9	DC13049	O
236390	2007-11-02 00:00:00.000	E03.9	DC13049	O
236390	2007-12-27 00:00:00.000	E03.9	DC13049	O
236390	2007-09-18 00:00:00.000	E03.9	DC13049	O
236390	2007-11-30 00:00:00.000	E03.9	DC13049	O
348221	2007-12-14 00:00:00.000	N40	DC9550	O
271645	1998-09-22 14:27:00.000	J03.9	DC4715	I

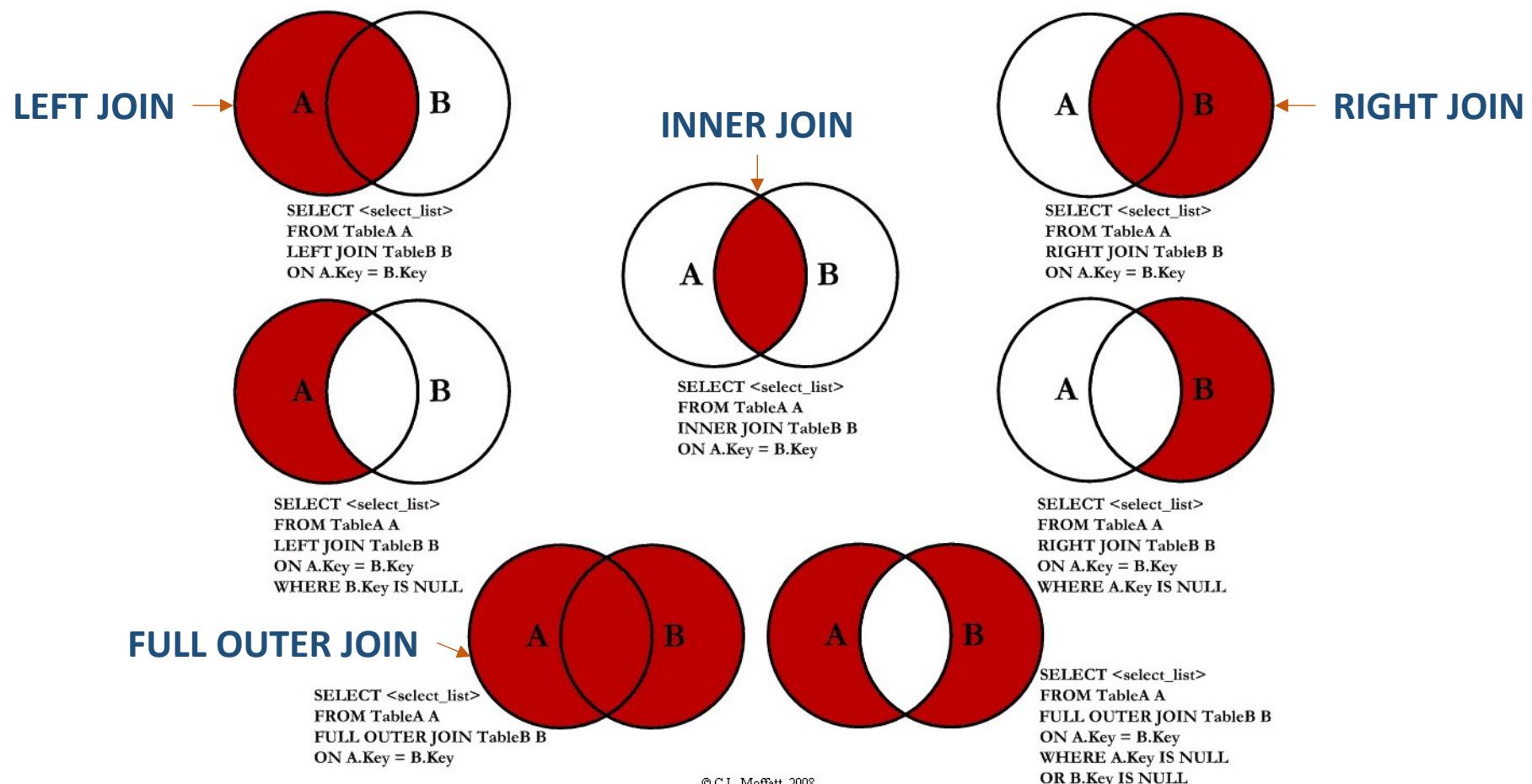
`select * from DiagnosisCodeMaster`

diaglocalcode	diagnosis
DC14574	Cyst of jaw NOS
DC14575	Other specified disorders of bone density and st...
DC14576	Gross hematuria
DC14577	Malignant neoplasm of extrahepatic bile duct
DC14578	Atrophic glossitis
DC14579	Ulceration of vagina
DC1458	Fracture of greater trochanter of femur, closed
DC14580	Perianal dermatitis
DC14581	Infection and inflammatory reaction due to prost...
DC14582	Sequelae of burn, corrosion and frostbite of trunk

personid	diagdate	diagcode	diaglocalcode	diagdept	diagnosis
118568	2001-07-29 16:39:00.000	R61.9	DC9940	I	Hyperhidrosis, unspecified
391675	1999-11-18 15:43:00.000	C22.0B	DC13882	I	Hepatoma
236390	2007-12-27 00:00:00.000	K21.0	DC13744	O	Gastroesophageal reflux disease with esophagitis
236390	2007-10-04 00:00:00.000	E03.9	DC13049	O	Other hypothyroidism
236390	2007-11-02 00:00:00.000	E03.9	DC13049	O	Other hypothyroidism
236390	2007-12-27 00:00:00.000	E03.9	DC13049	O	Other hypothyroidism
236390	2007-09-18 00:00:00.000	E03.9	DC13049	O	Other hypothyroidism
236390	2007-11-30 00:00:00.000	E03.9	DC13049	O	Other hypothyroidism

# 조인(Join)

- Join type



# 조인(Join)

명령	문법	설명
일반적인 조인	<pre>SELECT &lt;속성들&gt; FROM 테이블1, 테이블2 WHERE &lt;조인조건&gt; AND &lt;검색조건&gt;</pre>	SQL 문에서는 주로 동등조인을 사용한다.
	<pre>SELECT &lt;속성들&gt; FROM 테이블1 INNER JOIN 테이블2 ON &lt;조인조건&gt; WHERE &lt;검색조건&gt;</pre>	두 가지 문법 중 하나를 사용할 수 있다.
외부조인	<pre>SELECT &lt;속성들&gt; FROM 테이블1 {LEFT  RIGHT  FULL [OUTER]} JOIN 테이블2 ON &lt;조인조건&gt; WHERE &lt;검색조건&gt;</pre>	외부조인은 FROM 절에 조인 종류를 적고 ON을 이용하여 조인조건을 명시한다.

# 부속질의

명칭	위치	영문 및 동의어	설명
스칼라 부속질의	SELECT 절	scalar subquery	SELECT 절에서 사용되며 단일 값을 반환하기 때문에 스칼라 부속질의라고 한다.
인라인 뷰	FROM 절	inline view, table subquery	FROM 절에서 결과를 뷰(view) 형태로 반환하기 때문에 인라인 뷰라고 한다.
중첩질의	WHERE 절	nested subquery, predicate subquery	WHERE 절에 술어와 같이 사용되며 결과를 한정시키기 위해 사용된다. 상관 혹은 비상관 형태다.

# 부속질의-스칼라 부속질의

- 스칼라 부속질의(scalar subquery)란?
  - 부속질의의 결과 값을 단일 행, 단일 열의 스칼라 값으로 반환
  - 원칙적으로 스칼라 값이 들어갈 수 있는 모든 곳에 사용 가능
  - 일반적으로 **SELECT 문과 UPDATE SET 절에 사용**
  - 주질의와 부속질의와의 관계는 상관/비상관 모두 가능

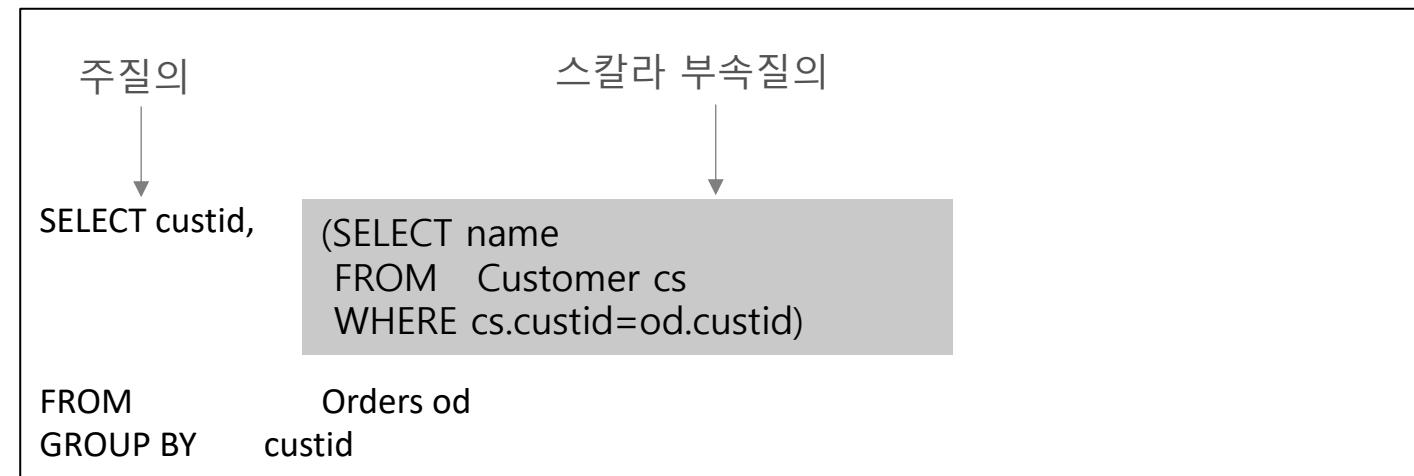


그림 4-3 스칼라 부속질의

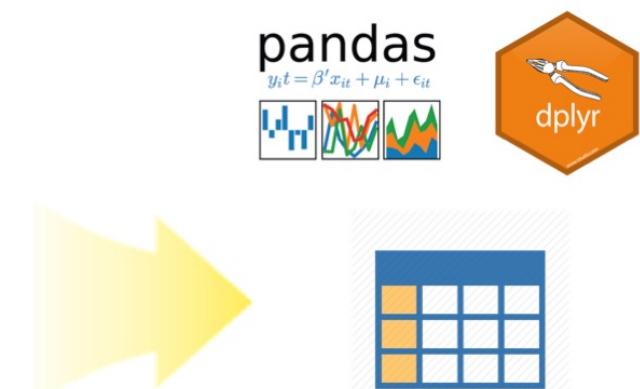
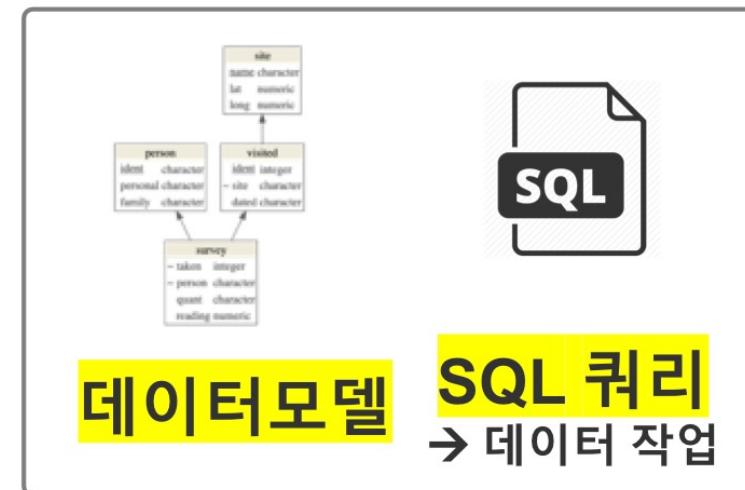
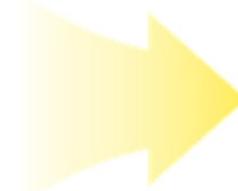
# RDB and R/Python



PostgreSQL



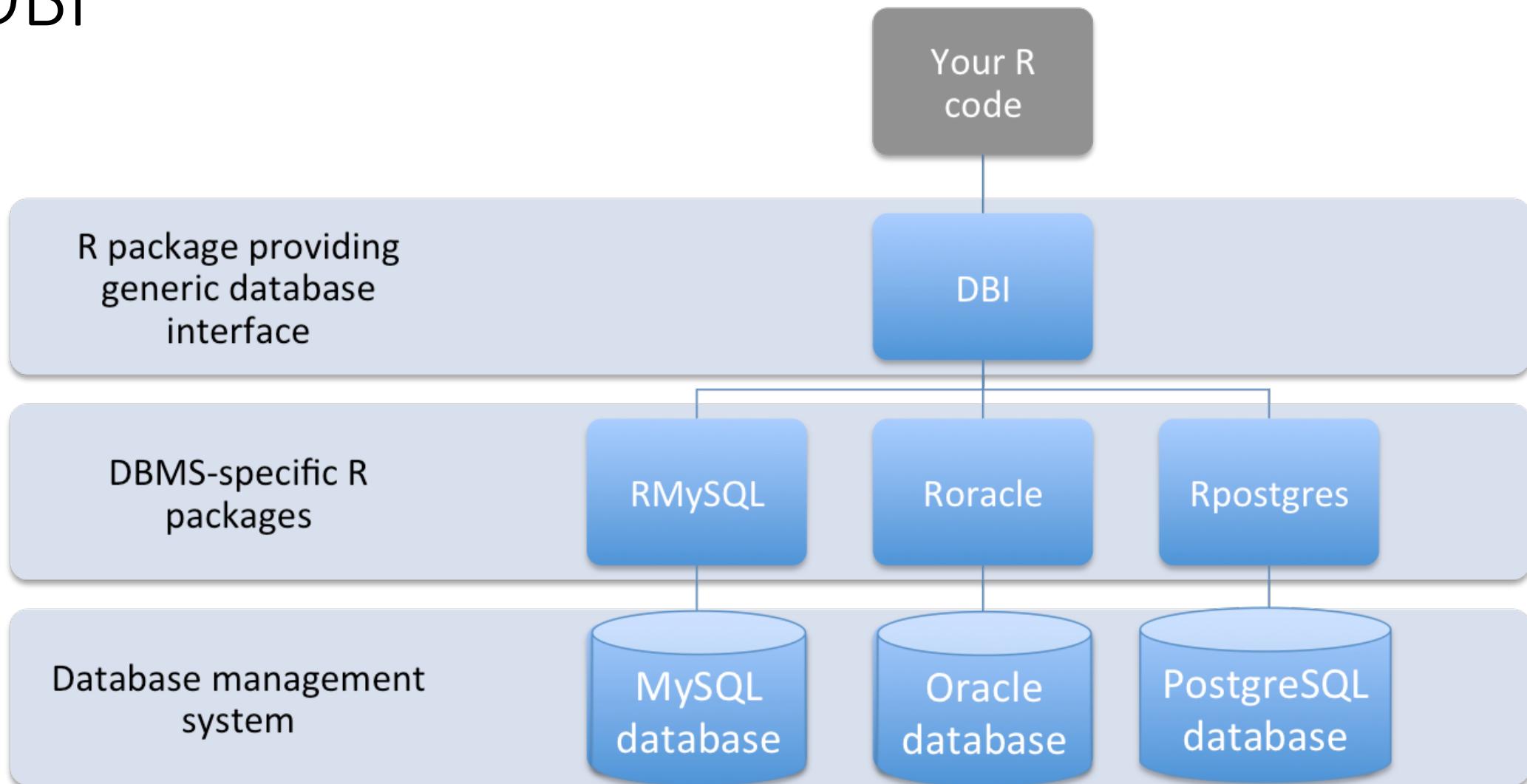
Google BigQuery



데이터프레임

<https://statklee.github.io/data-science/ds-database-dbi.html>

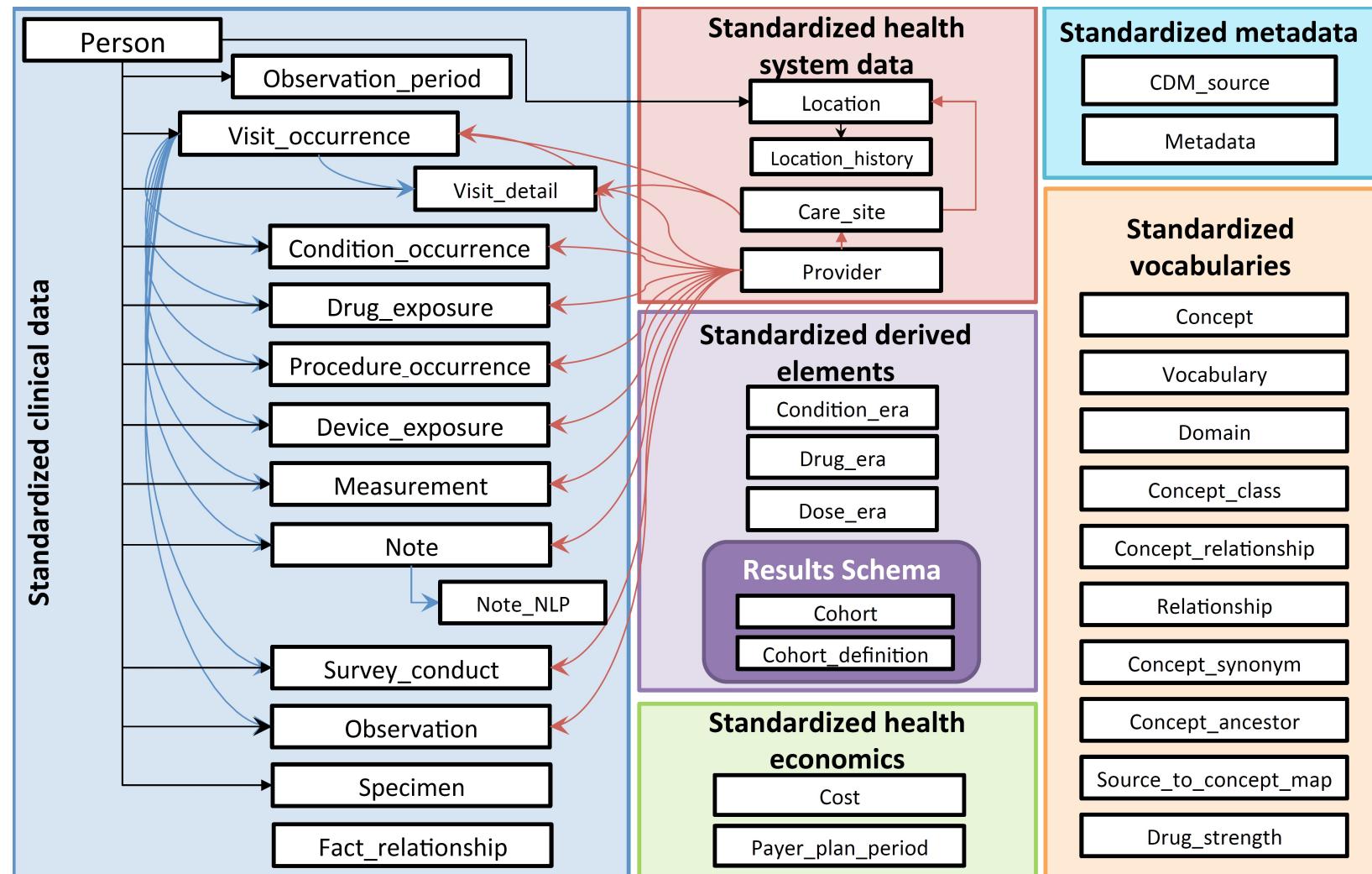
# DBI



# DBI 함수

DBI 함수	기능 설명
dbDriver	데이터베이스 인스턴스에 대한 드라이버 객체 생성
dbConnect	데이터베이스 인스턴스에 연결
dbListTables	연결된 데이터베이스 인스턴스에 포함된 테이블 목록을 출력
dbListFields	해당 테이블 내부 필드 목록을 출력
dbSendQuery	연결된 데이터베이스 인스턴스에 쿼리를 전달
dbFetch	SQL 쿼리 결과를 R 데이터프레임으로 가져옴
dbGetQuery	데이터베이스 인스턴스로부터 데이터를 쿼리함과 동시에 가져옴
dbClearResult	SQL 쿼리로 가져온 결과(result set)를 닫음
dbWriteTable	신규 테이블을 데이터베이스 인스턴스에 기록함
dbRemoveTable	데이터베이스 인스턴스에서 테이블을 삭제함
dbDisconnect	데이터베이스 인스턴스에 연결을 해제함

# 공통 데이터 모델 (OMOP-CDM)



# OMOP-CDM 필드 명명 규칙

명명	설명
[Event]_ID	테이블 별 각 행의 고유 식별자로, 테이블 간 관계를 설정하는 외래 키 역할을 한다. 예를 들어 PERSON_ID는 각 개인을 고유하게 식별한다. VISIT_OCCURRENCE_ID는 방문을 고유하게 식별한다.
[Event]_CONCEPT_ID	CONCEPT 참고 테이블의 standard concept에 대한 외래 키. 이는 모든 분석에 기반이 되는 사건의 주요 표현이다. 이 필드에는 OMOP 용어집에 정의된 concept ID 만을 사용할 수 있다. 예를 들어 CONDITION_CONCEPT_ID = 31967에는 SNOMED concept인 “오심 Nausea”에 대한 참조 값을 포함하고 있다.

# OMOP-CDM: Person

- 그녀는 36세 여성이다
- 그녀의 생년월일은 1982년 3월 12일이다
- 그녀는 백인이다
- 그녀는 영국인이다

이를 염두에 두면 PERSON 테이블을 다음과 같이 나타낼 수 있다:

Table 4.3: PERSON 테이블.

필드 Column 이름	값	설명
PERSON_ID	1	PERSON_ID는 원천에서 직접적으로 생성되거나 변환 과정의 일부분으로 생성된 정수여야 한다.
GENDER_CONCEPT_ID	8532	여성을 의미하는 concept ID는 8532이다.
YEAR_OF_BIRTH	1982	
MONTH_OF_BIRTH	3	
DAY_OF_BIRTH	12	

필드 Column 이름	값	설명
BIRTH_DATETIME	1982-03-12 00:00:00	시간을 정확히 알 수 없는 경우 자정으로 한다.
DEATH_DATETIME		
RACE_CONCEPT_ID	8527	백인을 의미하는 concept ID는 8527이다. 영국인이라는 race는 4093769이다. 둘 다 해당할 경우 후자를 활용한다. 민족성은 ETHNICITY_CONCEPT_ID가 아닌 race의 일부로써 여기에 저장된다는 점에 유의하라.
ETHNICITY_CONCEPT_ID	38003564	이는 히스패닉을 다른 사람과 구분하기 위해 사용되는 전형적인 미국식 표기법이다. 이 경우 영국인이라는 민족성은 RACE_CONCEPT_ID에 저장된다. 미국 이외의 지역에서는 사용되지 않는다. 38003564는 “히스패닉이 아님”을 나타낸다.
LOCATION_ID		주소는 알려지지 않았다.
PROVIDER_ID		일차 진료 제공자는 알려지지 않았다.
CARE_SITE		일차 진료 장소는 알려지지 않았다.

# OMOP-CDM: Condition\_occurrence

월경통이라고 하는 고통스러운 월경 경련의 SNOMED 코드는 266599000이다. 표 4.7은 CONDITION\_OCCURRENCE 테이블에 어떻게 표시되는지를 보여준다:

Table 4.7: CONDITION\_OCCURRENCE 테이블.

필드 이름	값	설명
CONDITION_OCCURRENCE_ID	964	이는 일반적으로 테이블의 각 기록에 대한 고유 식별자로서 자동으로 생성되는 값이다.
PERSON_ID	1	이는 PERSON 테이블에서 Laura의 기록에 대한 외래 키이며 PERSON을 CONDITION_OCCURRENCE에 연결한다.
CONDITION_CONCEPT_ID	194696	SNOMED 코드 266599000을 나타내는 외래 키 (concept ID): 194696.
CONDITION_START_DATE	2010-01-06	Condition의 인스턴스가 기록된 날짜이다.
CONDITION_START_DATETIME	2010-01-06 00:00:00	Condition의 인스턴스가 기록된 날짜 및 시간이다. 시간을 알 수 없으므로 자정으로 입력한다.
CONDITION_END_DATE	NULL	이는 인스턴스가 종료된 것으로 여겨지는 날짜지만 거의 기록되지 않는다.



Thank you